# Automatic Extraction of I/O Data for Scripted Benchmarks

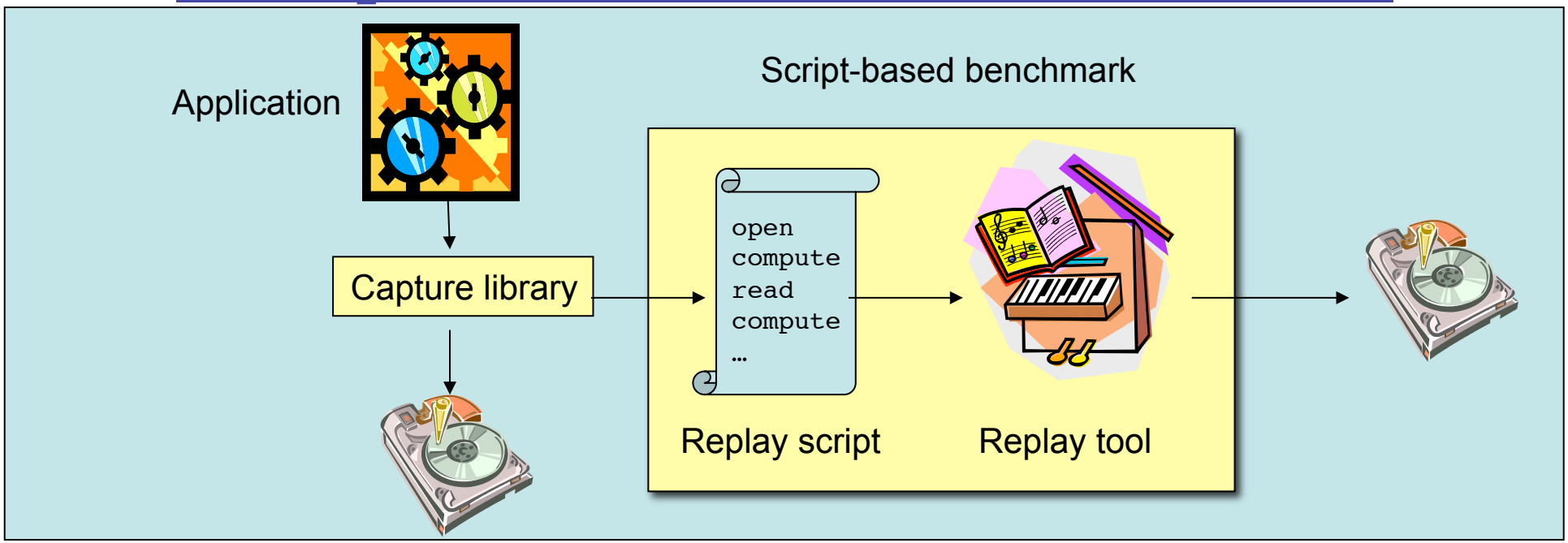
John May

Lawrence Livermore National Laboratory

# Standard I/O Benchmarks

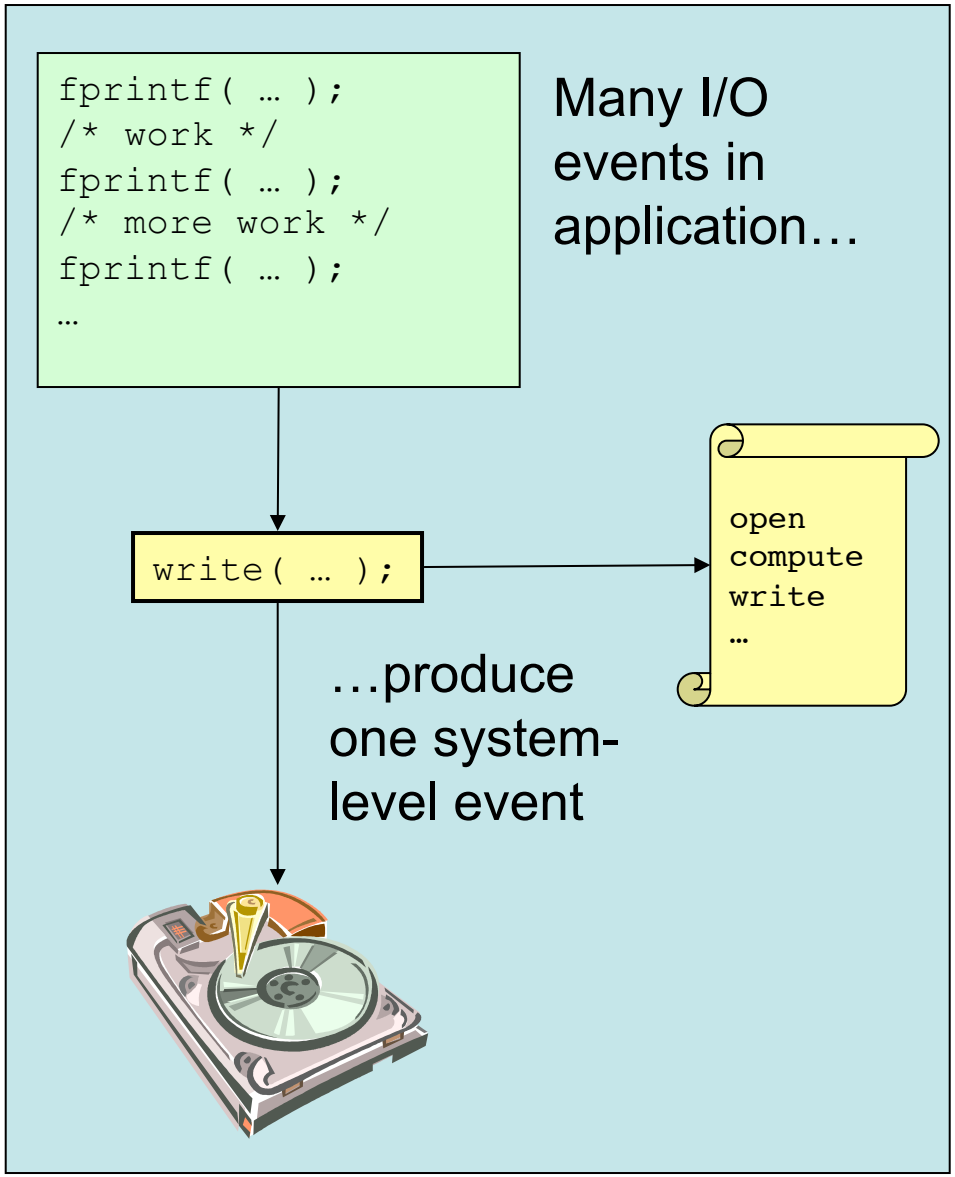
- Application-based benchmarks
  - Use I/O system in realistic ways
  - May be hard to build, run, or distribute
- Synthetic benchmarks
  - Use standard or custom I/O patterns
  - May not present a realistic workload

# Script-Based Benchmarks




- Record I/O events in application
- Replay events and timing
  - Based on //TRACE by Mesnier et al. [1]
  - Our current focus is *sequential* I/O

# Scripted Benchmark Challenges

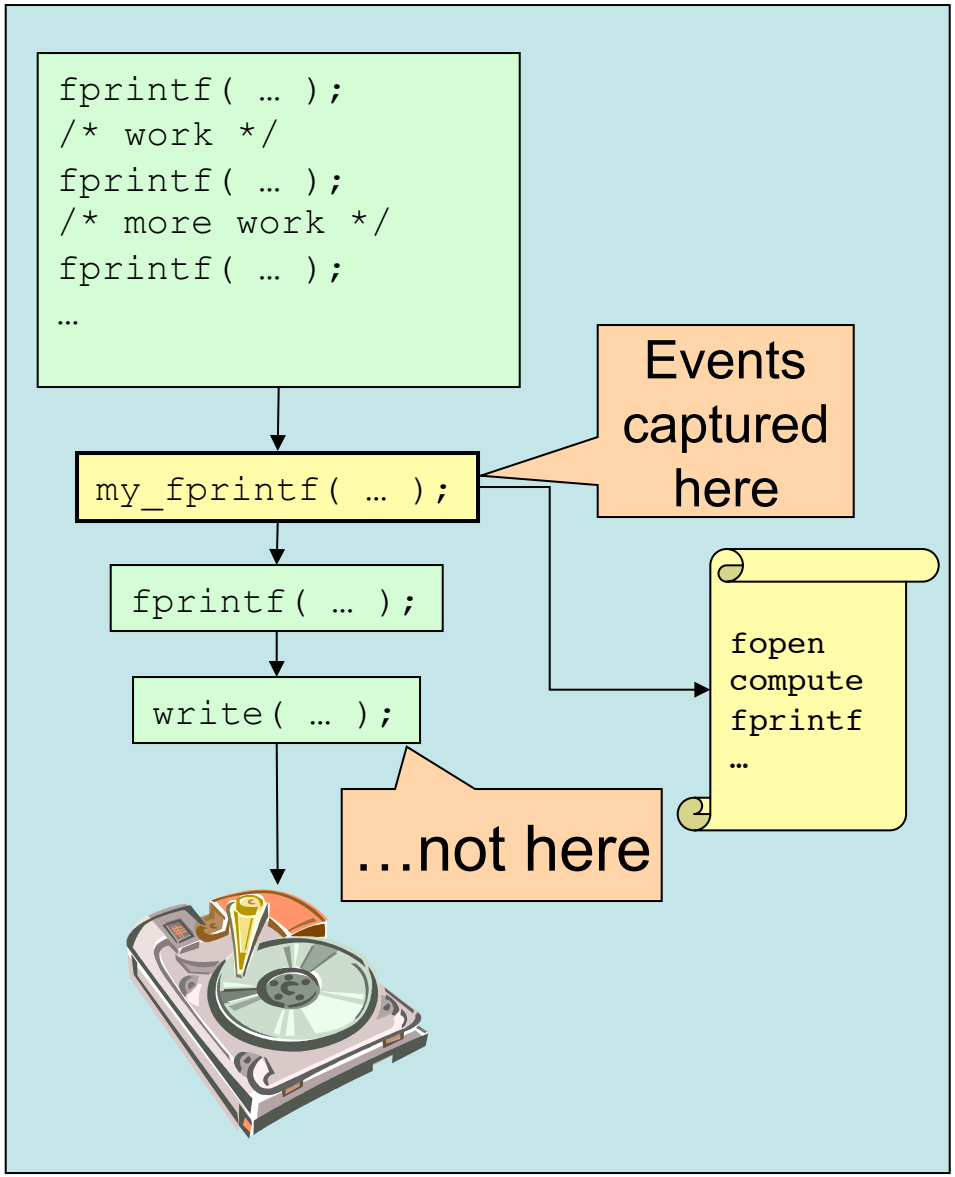



- Instrument events at the right level
- Gather accurate timing data

- Accurate replay
  - not covered here

# Instrumentation options

# Library Interposition




- Intercept standard I/O library functions
- Insert instrumented versions
- Wrong level: can't capture low-level calls from high-level functions

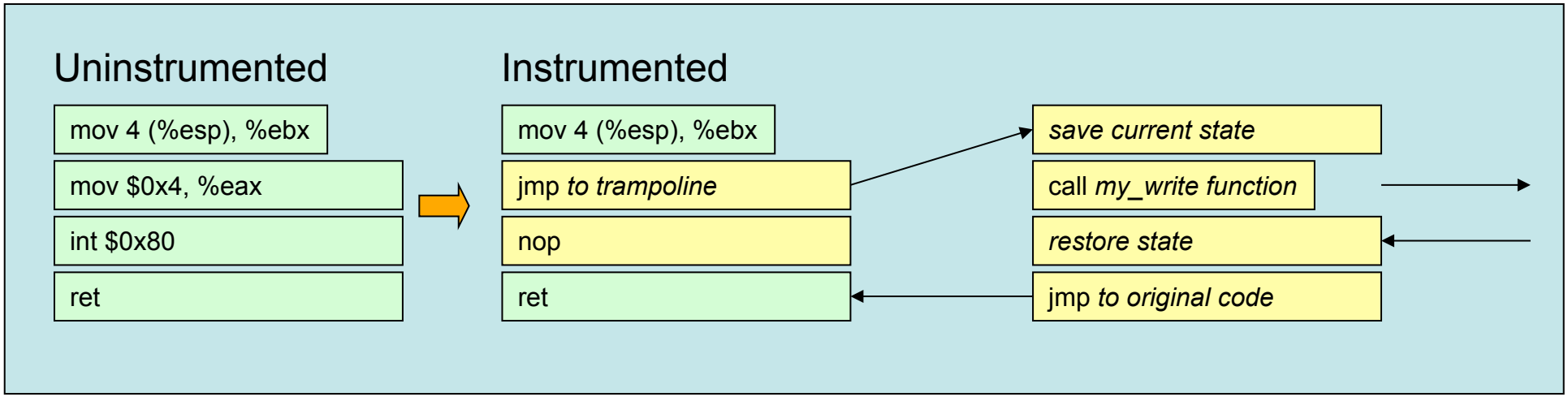
# Linux Strace Utility

```
$ strace -r -T -s 0 -e trace=file,desc ls
     0.000000 execve("/bin/ls", [, ...], [/* 29 vars */]) = 0 <0.000237>
     0.000297 open("/etc/ld.so.cache", O_RDONLY) = 3 <0.000047>
     0.000257 fstat64(3, {st_mode=S_IFREG|0644, st_size=64677, ...}) = 0 <0.000033>
     0.000394 close(3)                       = 0 <0.000015>
     0.000230 open("/lib/librt.so.1", O_RDONLY) = 3 <0.000046>
     0.000289 read(3, ""..., 512)            = 512 <0.000028>
     ...
```

Inter-event times exaggerated

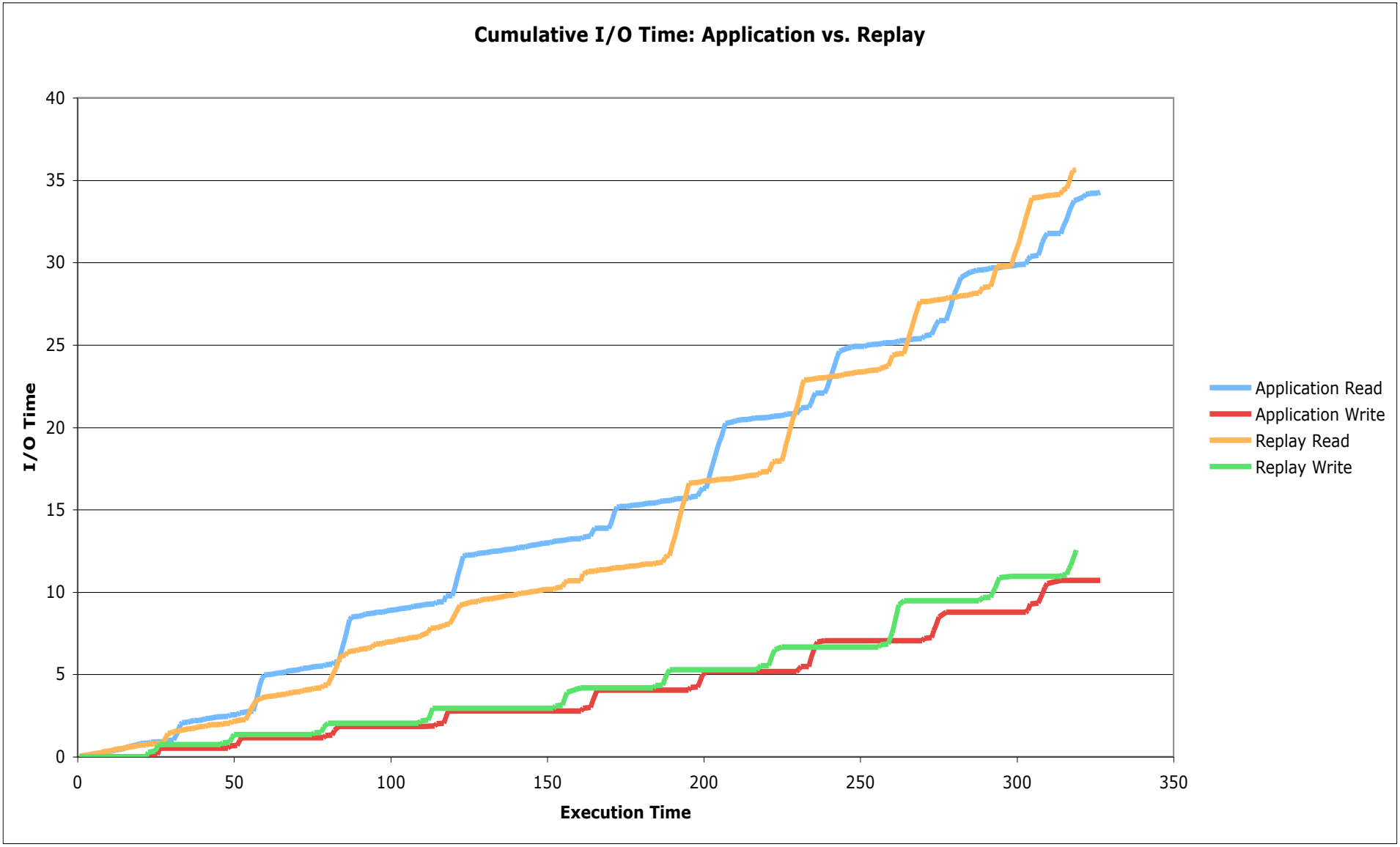
- Tracks system calls using debugger interface
- Easy to use; output easy parse
- Adds too much overhead to “compute” times

# Dynamic Binary Instrumentation




- System call instructions replaced with calls to user instrumentation
- Low overhead, accurate timing
- Instrumentation limited to 32-bit Linux
- Based on Jockey library [2]

# Replay Profile vs. Application



# Acknowledgements and Citations

Thanks to Maya Gokhale, Roger Pearce, John Gyllenhaal, Mark Grondona, and Ben Woodard for help, discussions, and support.

[1] Mesnier, M., Wachs, M., Sambasivan, R. R. , Lopez, J., Hendricks, J., and Ganger, G. R., "//TRACE: Parallel trace replay with approximate causal events." In *Proceedings of the 5th USENIX Conference on File and Storage Technologies* (FAST '07) (February 2007).

[2] Saito, Y., "Jockey: A user-space library for record-replay debugging." In *AADEBUG '05: Proceedings of the Sixth International Symposium on Automated Analysis-Driven Debugging* (2005), pp.69–76.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344.